"Application Program Interface for Transforming Heterogeneous Programs" Application Serial No.: 09/343,276; and "Shared Library Optimization for Heterogeneous Programs" Application Serial No.: 09/343,279. Each of the above applications were filed on the same day as the present application and assigned to the same assignee.

Please replace the paragraph beginning on page 19, line 18, and ending on page 20, line 2, with the following paragraph:

The binary for a component is obtained (block 401) and analyzed (block 403). The process of analyzing a binary to discern its code blocks and data blocks is often referred to as "code discovery." An exemplary embodiment of a code discovery operation 410 illustrated in FIG. 4B is described with reference to the methodology disclosed in U.S. Patent Number 5,664,191, assigned to the assignee of the present application. Various other code discovery methodologies are currently in use in the art and could be easily substituted by one of skill in the art in block 403.

Please replace the paragraph beginning on page 20, line 3, and ending on page 20, line 15, with the following paragraph:

In the exemplary embodiment, the PDB file 202 is assumed to contain entry points, export entry tables, jump tables, and symbol tables. A first approximation of the basic blocks is created at block 411 using the locations of procedures, labels, and data as defined in the PDB file 202. A bit map of the binary is created in which each bit represents an address in the binary. The beginning address of each block is marked in the bit map as either a code block or a data block. All entry points listed in the PDB file 202 are next marked in the bit map (block 413). The entry points are assumed to mark the beginning of a code block. At this point in the process, the length of a block is assumed to extend from its beginning mark in the bit map until another mark is encountered. However, a block marked as data can contain code that does not have an entry point. Because the component's creator knows where the various blocks begin and end, the process checks to see if a user input command file was provided (block 415) and uses it, if available, to override any default designation of a code block as data (block 417).